Use of CLIPS for Representation and Inference in a Clinical Event Monitor

Marvin C. Pankaskie and Michael M. Wagner
Center for Biomedical Informatics, University of Pittsburgh

We developed a clinical event monitor that is currently deployed in an inpatient setting. We selected CLIPS as the basis for its KB and inference engine. This paper describes the considerations that went into that decision, how we represented drug and laboratory knowledge in CLIPSs, and how we extended CLIPS to deal with temporal inferences We also review the published literature about the use of CLIPS in medicine.

INTRODUCTION

A key decision faced by developers of clinical event monitors is the choice of representation and inference subsystems [1]. This decision affects the ease of knowledge acquisition, representation, and the range of inferences that the system can make. In developing a clinical event monitor at the University of Pittsburgh Medical Center (UPMC), we initially wrote rules using the C programming language because issues of data access, embedding and communication of alerts to physicians were paramount. As we attempted to add rules to the system, the use of a knowledge representation system became necessary.

Knowledge representation (KR) systems are complex software systems and many exist. Therefore, we did not consider building one ourselves. Instead, we considered general purpose KR services such as the C Language Integrated Production System (CLIPS), and special purpose KR systems such as Arden.

INFERENCE ENGINES USED IN CLINICAL EVENT MONITORING SYSTEMS

Most of the inference engines (IE) employed in clinical decision support are either specially designed systems such as Arden or general purpose systems such as CLIPS, and each has its own unique set of characteristics. They are designed to carry out some reasoning process, using sets of clinical criteria or "rules" depending on the event that triggered the IE. If the criteria are met, then the IE provides some action (or choice of actions) to alert the provider that a clinical event of interest has occurred and that some action(s) should be taken. Notification can take any number of forms, but email, paging, updating of a clinical database, and quality assurance have been the most common to date. Several examples of systems currently in use will be summarized below.

The Arden Syntax for Medical Logic Modules is a special purpose language for encoding medical knowledge, developed in part at Columbia University [2]. It consists of a Medical Logic Module (MLM), which contains sufficient logic for a single medical decision; MLMs are therefore independent of each other. Functionally, an MLM consists of four major components: an evoking event, logic, action, and data mapping. Since all necessary knowledge is contained within the MLM, there is no need for an external knowledge base. While this feature can enhance processing efficiency (only data needed by the logic component is retrieved), knowledge cannot be shared between MLMs (e.g., drug product and pharmacological profile data). The primary advantages to using Arden are that it includes a well developed concept of time, and thus temporal inferences can be encoded more easily, and the rule logic can be shared among users with only a minimal amount of recoding. Arden Syntax has also been incorporated into several commercial products such as the Clinical Decision Support/CareVISION system by HealthVISION Corporation [3] and the Clinical Management System being developed by IBM.

Stanford has developed a special purpose software system called EON as the inference component or "problem solver" of their protocol-based oncology care systems, PROTÉGÉ II and T-HELPER. Using a problem-solving approach, PROTÉGÉ II allows a developer to assemble new problem-solving methods and models, using a library of reuseable methods and mechanisms [4].

Since 1972, the Health Evaluation through Logical Processing (HELP™) system has been used at the LDS Hospital for clinical decision support, knowledge acquisition and database development, application development, and systems integration. It is largely constructed using the Tandem system language (TAL), a high-level database query language (PAL) the HELP frame language, and the HELP compiler language (HCOM), the latter serving to facilitate the development of computerized logic modules or sectors. HELP™ is now a trademarked product of 3M Health Information Systems, and versions of it are commercially available [5].

Brigham and Women's Hospital employs an application suite written in MUMPS with a knowledge base stored in a distributed MUMPS database, using relevant patient information from the Brigham Integrated Computing System (BICS). To process asynchronous events, the BICS Event Engine was developed to facilitate the development and production use of clinical logic rules. A rule editor provides a simple interface so that rules can be authored by non-programmers. Although the Event Engine's triggering, alerting, and action-item functions are broader than those included in the Arden Syntax, the logic of the Event Engine's rules are compatible with those described by Arden [6].

The Regenstrief Medical Record System (RMRS) uses an English-like computer language system called CARE to specify the retrieval of patient information according to criteria set by the user. It can be used in surveillance mode to monitor events and procedures, invoke a set of practice rules (protocols), and transmit its inferences to the physician concerning clinical conditions that might require corrective action. It can also be used in query mode to extract and analyze patient characteristics or care patterns (e.g., the incidence of toxicity for a new drug or usage trends for a particular diagnostic test). The latter can be used for both retrospective analysis or for incorporation into prospective studies [7].

GermWatcher, developed at Washington University's School of Medicine, is an expert system designed to support infection control specialists in detecting, tracking, and investigating infections in hospitalized patients. It employs a commercial relational database to model the CDS's National Nosocomial Infection Surveillance System culture-based definitions for nosocomial infections plus the Germ Watcher Engine, a modified implementation of the CLIPS expert system shell. The concept of GermWatcher was first described and implemented at the LDS Hospital at the University of Utah, but was non-portable and employed a self-developed database and inference engine. The WU implementation added a family of tools using readily available and public-domain products, including the CLIPS inference engine [8].

CLIPS is a forward chaining, multiparadigm, expert system shell that provides support for rule-based, object-oriented, and procedural programming. CLIPS rules closely resemble those found in languages such as ART, ART-IM, Eclipse, and Cognate, while its programming language has features similar to languages such as C, Ada, and Pascal, and is syntactically very similar to LISP. CLIPS was designed at NASA's Johnson Space Center, is written in the C programming language, and is available in the public domain [9]. It can be installed on a variety of computers, and versions are available for the PC, Mac, and UNIX operating systems. Key features of CLIPS that made it attractive to us were that it supports object-oriented programming, provides structures for taxonomic inferences (e.g., to classify a medication's pharmacological properties), supports both embedding and arbitrary user extension through the use of external functions, and supports the construction of modules for more explicit execution control and knowledge base partitioning. Its primary drawback is the lack of temporal functions such as those found in Arden (e.g., retrieving the last 5 serum K levels, detecting a medication order placed within 24 hrs of an alert, testing the existence of a culture report).

OVERVIEW OF THE CLINICAL EVENT MONITORING SYSTEM (CLEM) AT UPMC

Out clinical event monitoring system (CLEM) employs an orthodox architecture for an event monitor [1]. It consists of an event detector, an inference engine, knowledge bases, performance monitors, coverage list database, and notification subsystems. It runs under the Unix operating system and obtains patient data from the existing central data warehouse at UPMC (MARS). It employs CLIPS as both the inference engine and knowledge base, which are embedded in a structured "manager" program written in the C programming language. Notification is made via email or two-way pager, depending on the urgency of the information. Additional documentation related to the event (e.g., therapeutic guidelines, literature reviews) can be sent to the recipient on demand via email attachments, direct printouts to nursing station printers, or as HTML pages.

Examples of the conditions or events which CLEM monitors include changing renal function in patients receiving renally excreted medications, appropriate use of drug level monitoring, drug_induced platelet toxicity, life-threatening electrolyte abnormalities, medications that interact with warfarin, IV to oral administration route conversions, and optimal medication management in geriatric patients. The detection of these conditions poses the full-range of difficulty for expert systems including temporal and taxonomic inferences.

RESULTS AND OBSERVATIONS

1. Representation of Drug Knowledge

UPMC Pharmacy Services currently uses a commercial system (Pharmakon®) to enter and track medication orders and to alert the pharmacist to such potential events as drug allergies, drug order duplication, and potential drug interactions prior to the actual administration of the medication to the patient. Pharmakon records are usually limited to information about what formulary drug products are dispensed to the units based on paper orders written by the physician. rather than a medication administration record (i.e., drug name, dose, route, infusion rate). In addition, each formulary entry is assigned one or more arbitrary keys (mnemonic) that describe the strength, concentration, and dosage form of the product, and are used for internal identification and billing purposes. This key is similar to the National Drug Code, but has been customized to work with the Pharmakon system and is not readily mappable to other common drug codes such as National Drug Code (FDA), Generic Code Number (FirstData Bank) or Generic Product Identifier (Medispan).

In order for us to represent drug knowledge (e.g., default dosing schedules, side effects, pharmacological class membership, drug interactions), we developed a drug knowledge base, written as a series of CLIPS deffact statements, that includes fields for the generic drug name, mnemonic codes that relate to that drug, side effects, dosing information, pharmacologic and therapeutic class information, and drug interaction lists. An example of one entry for ciprofloxacin is given below:

```
(Drug

  (mnemonic CPRF250 CPRF500 CPRF750)
  (name Ciprofloxacin)
  (excretion_route renal)
  (therapeutic_class antibiotic)
  (pharmacologic_class fluoroquinolone)
  (side_effects  delirium diarrhea seizures)
  (clearance_dependant_doses
     "0.25 0.50 Q12H BID"
     "0.25 0.50 Q12H BID"
     "0.25 0.50 Q12H Q24H BID QD")
  (warfarin_interaction "Probably Potentiates"))
```

This declarative, frame-like representation has several advantages. First, it allowed us to easily add new attributes to existing entries (e.g., drug interactions, therapeutic restrictions, warnings, educational "pearls") without changing the rest of the knowledge base. Second, we were able to add new formulary drugs and new dosage forms (new strengths, extemporaneous or custom preparations) that are continually being added to the drug formulary (~ 5 per week).

2. Forward Chaining

A common inference in event monitoring is the determination of class membership (e.g., gentamicin is a member of the aminoglycoside class of antibiotics, and the aminoglycosides as a class can cause nephrotoxicity). Backward chaining is not available in CLIPS. After writing many rules that had a pseudo-backward chaining flavor, we realized that forward chaining key drug properties was advantageous. This allowed us to create for each patient a description of his/her current medications (drug name, dose, route and frequency of administration) as well as various properties (side effects, contraindications, drug interactions, dosing schedules) of each of these current medications as a single database record that could then be used by our inference engine. By combining patient-specific medication data with the drug knowledge described above, our forward-chaining rule asserts a set of facts as shown below for each medication the patient is on:

```
(patient_medication_data

  (mnemonic CPRF500)
  (name Ciprofloxacin)
  (dosage_form Ciprofloxacin HCl 500 mg tablet)
  (dose 500mg)
  (route oral)
  (frequency BID)
  (excretion_route renal)
  (therapeutic_class antibiotic)
  (pharmacologic_class fluroquinolone)
  (side_effects  delirium diarrhea seizures)
  (clearance_dependant_doses
     "0.25 0.50 Q12H BID"
     "0.25 0.50 Q12H BID"
     "0.25 0.50 Q12H BID Q24H QD")
  (warfarin_interaction "Probably Potentiates"))
```

3. Temporal Inferences

Another common type of inference is temporal inference. CLIPS provides good list processing primitives (e.g., nth) so if we loaded time sorted data into CLIPS, we could obtain sublists (e.g., a temporal representation of a patients medication orders or a table or graph of the patient's laboratory data sorted by date and time) with reasonable efficiency. However, we were concerned that some data that we would process would have numbers of observations that exceeded CLIPS capacity. Therefore, we chose to represent encoded data using a combination of linked lists and temporal predicates such as *last*, *previous*, *first_instance_of_event_after, timeof,* and *existence_of* to retrieve only events of interest. We also developed several storage databases that allowed us to track event details and message content back in time (and thus construct rules that were "aware" of message content that had already been sent) as well as to post messages for delivery at some predetermined time in the future. The development of these external functions required a significant amount of time compared to the construction of the drug knowledge base and the embedding of CLIPS in our C language "manager". However, coupled with the archival information available from our electronic medical record system, this design permitted us to create rules and alerts that utilized all available data, both past and present, and to more accurately predict possible events in the future (e.g., adverse drug reactions)

4. Automatic Outcome Detection

To facilitate gathering information on whether the recipient heeded or ignored the alert's advice, and what actions were taken (if any), we constructed a set of automated outcome detection (AOD) rules. When a given alert is sent, information such as the triggering event that lead to the alert, the time that the alert was sent, and the recipient(s) of the alert is stored in a database that can be loaded into CLIPS working memory the next time that CLIPS initializes. This information can then used to infer if any action has been taken relative to the time/content of the original alert (e.g., if a K supplement was ordered if the serum K level was low or if a serum phenytoin level was ordered if the patient had been on phenytoin for more than 3 days), using a series of rules that compare the patient's status before and after the alert was sent. In this manner, we are able to observe whether the recipient had, in fact, changed his/her management of the patient.

DISCUSSION

Currently, the drug KB has 450 drug formulary entries, each entered by hand using information from the UPMC Drug Formulary, various literature reviews, current practice guidelines, and electronic drug information references (Micromedex, Medispan, Clinical Pharmacology). This is approximately 10% of the total UPMC Drug Formulary listing. While CLIPS provides an excellent environment for building a drug KB and using it for taxonomic inferences, representing and maintaining the entire drug formulary in this manner would be difficult. Yet some linkage between Pharmakon and our drug knowledge base must be maintained to accurately identify the drug product dispensed by Pharmacy Services. To resolve this issue, we intend to obtain a commercial drug database that would serve as the drug knowledge base component of CLEM, and create a mapping of the Pharmakon mnemonic key(s) to the appropriate unique identifier key or NDC code in the commercial database. This will provide greater portability, improve database maintenance, and accommodate changes to (or replacement of) the Pharmakon database in the future.

We currently represent temporal elements (e.g., previous serum K level, last serum creatinine concentration, a new order for vancomycin) through the use of external C functions that pass values to CLIPS. To extend the temporal reasoning ability of CLEM, we are considering converting the rules built in CLIPS into Medical Language Modules (MLMs) using the Arden Syntax language, which has better built-in date and time functions.

Representing practice guidelines or critical care pathways as rules also poses some unique challenges. Starren and Xie [10] compared three knowledge representation formalisms (CLASSIC, PROLOG, and CLIPS) used to encode the National Cholesterol Education Program (NCEP) guidelines. They found that CLIPS was the easiest system to conceptualize states and rules and to specify patient attributes, although it lacked built-in time and date functions. The Computer Assisted Management Protocol (CAMP) developed at Duke University applied individualized feedback reminder techniques to patients with diabetes mellitus and demonstrated major improvement in guideline compliance [11]. We currently incorporate several practice guidelines (e.g., antibiotic dosing in patients with renal impairment, switching IV antibiotics to their oral equivalents, optimal utilization of amphotericin B in *Candida sp.* infections) into CLEM, and intend to add others in the future.

CONCLUSIONS AND FUTURE DIRECTIONS

We have developed a clinical event monitor (CLEM) built around an existing electronic medical record system (MARS), using a combination of an embedded CLIPS-based inference engine and drug knowledge base, a C language event detector and notification manager, and an email-based messaging service. CLEM is currently evaluating events such as abnormal blood chemistries (e.g., hematocrit, serum creatinine, electrolyte levels, INR), drug dosing in patients with impaired renal function (e.g., antibiotics, narcotics, H_2 blockers), drug use in geriatric patients (e.g., excessive use of sedatives, anticholinergic side effects), and medication management initiatives (e.g., drug level monitoring, IV to oral route conversion, warfarin interactions, amphotericin therapy). Information concerning events of clinical importance or suggestions for improving drug therapy outcomes are currently sent to general medicine interns via email or two-way pagers, and they responded favorably to them. Changes in patient management are tracked using automated outcome detection rules and manual chart reviews.

Future refinements to CLEM will involve exporting our CLIPS-based rule set to Arden Syntax (providing we can retain the inference capabilities we now have using CLIPS) to allow the development of better temporal inferences, the purchase of a commercial drug knowledge base that will be easier for us to upgrade and maintain, and the development of a mapping algorithm to link the current Pharmacy Services medication ordering and tracking system with the commercial drug database.

This work was supported by grants 5 T15 LM07059-10 and 1 R29 LM06233-01A1 from the National Library of Medicine.

REFERENCES

1. Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a Clinical Event Monitor. *Computers and Biomedical Research* 1996; 29(3):194-221.

2. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. *Computers and Biomedical Research* 1994; 27(4):291-324.

3. Corman R, Sager J, Wu J, Wang S, Diane D, McEwan D. Arden-Based Clinical Decision Support in an Object Oriented World. *Proc AMIA Annual Fall Symposium* 1996; 873.

4. Musen MA, Gennari JH, Eriksson H, Tu SW, Puerta AR. PROTÉGÉ-II: Computer Support for Development of Intelligent Systems from Libraries of Components. *Proc of MEDINFO 95, The Eighth World Congress on Medical Informatics* 1995; 766-770.

5. Kuperman GJ, Gardner RM, Pryor TA, **HELP: A Dynamic Hospital Information System**. Springer-Verlag 1991.

6. Kuperman GJ, Teich JM, Bates DW, McLatchey J, Hoff TG, Representing Hospital Events as Complex Conditionals. *JAMIA* 1995; 2:137-141.

7. McDonald CJ, Tierney W, Martin DK, Overhage JM, The Regenstrief Medical Record System: Twenty Years of Experience in Hospitals, Clinics, and Neighborhood Health Centers. *MD Comput* 1992; 9:206-217.

8. Kahn MG, Steib SA, Fraser VJ, Dunagan WC. An Expert System For Culture-Based Infection-Control Surveillance. *Proc 17th Annual Symposium on Computer Applications in Medical Care* 1993: 171-5.

9. The Information Technology Office, Johnson Space Center, National Aeronautics and Space Administration (NASA), *CLIPS: The C Language Integrated Production System.* Http://www.jsc.nasa.gov/~clips/CLIPS.html.

10. Starren J, Xie G. Comparison of Three Knowledge Representation Formalisms for Encoding the NCEP Cholesterol Guidelines. *Proc 18th Symp Comput Appl Med Care* 1994; 18:792-796.

11. Lobach DF, Hammond WE. Development and Evaluation of a Computer-Assisted Management Protocol (CAMP): Improved Compliance with Care Guidelines for Diabetes Mellitus. *Proc 18th Symp Comput Appl Med Care* 1994; 18:787-791.